



Procedia Computer Science

Volume 66, 2015, Pages 679–688

YSC 2015. 4th International Young Scientists Conference on
Computational Science

CAVE 3D: software extensions for scientific visualization of large-scale models

Natalia Melnikova^{1,3*}, Stepan Orlov¹, Nikolay Shabrov¹, Vlad Kiev¹,
Aleksy Kuzin¹, Michael Resch², Uwe Woessner² and Martin Aumüller²

¹*Peter the Great St. Petersburg Polytechnic University, Russia*

²*High Performance Computing Centre Stuttgart (HLRS), Germany*

³*ITMO University, Russia*

*naunat@mail.ru, majorsteve@mail.ru, shabrov@rwws.ru, nikesk@mail.ru,
kuzin_aleksei@mail.ru, resch@hlrs.de, woessner@hlrs.de, aumueller@hlrs.de*

Abstract

Numerical analysis of large-scale and multidisciplinary problems on high-performance computer systems is one of the main computational challenges of the 21st century. The amount of data processed in complex systems analyses approaches peta- and exascale. The technical possibility for real-time visualization, post-processing and analysis of large-scale models is extremely important for carrying out comprehensive numerical studies. Powerful visualization is going to play an important role in the future of large-scale models. In this paper, we describe several software extensions aimed to improve visualization performance for large-scale models and developed by our team for 3D virtual environment systems such as CAVEs and Powerwalls. These extensions include an algorithm for real-time generation of isosurfaces on large meshes and a visualization system designed for massively parallel computing environment. Besides, we describe an augmented reality system developed by the part of our team in Stuttgart.

Keywords: CAVE 3D, visualization, virtual and augmented reality, large-scale models, isosurface generation

1 Introduction

Numerical analysis of large-scale and multidisciplinary problems on high-performance computer systems is one of the main computational challenges of the 21st century. That is why large-scale computational challenges have been closely considered in the IESP Roadmap — a document which cardinally reviews software development strategies for the time period until 2019. A technical

* Corresponding author

possibility for real-time visualization, post-processing and analysis of large-scale models is extremely important for carrying out comprehensive numerical studies; thereby, powerful visualization is going to play an important role in the future of large-scale simulations.

Challenging problems often require peta- and exaflops computing applications for their solution. The amount of data processed in these problems is supposed to reach peta- and exascale, which requires implementing new software and hardware solutions. A powerful and advanced hardware-software solution for large-scale model visualization are projection based immersive real-time virtual environments (VEs) such as CAVEs (Cave Automatic Virtual Environment), Powerwalls or dome projections. CAVEs are nowadays widely used in research institutions and industrial companies working in the fields of aerospace, aviation, automotive industry and shipbuilding, in studies of complex physical phenomena in gas and fluid dynamics, chemistry, biology, combustion and earthquake simulations.

At present, leading universities of the world (including Iowa State University, University of California in San Diego, King Abdullah University of Science and Technology in Saudi Arabia, Universities of Stuttgart, Aachen and Paris-Sud) are upgrading their virtual environment to systems with resolution of up to 200 megapixels. The largest 5-sided CAVE in Europe has been installed at the University of Aachen: the size of its side wall display is 5.25 m \times 3.30 m. The growing resolutions of VE systems further increase computational costs on scientific visualization.

In order to reduce computational costs on large-scale models visualization, we have designed several software tools, described below in this paper. These tools include: (a) implementation of an original algorithm for generating isosurfaces on a large mesh in real-time (described in Section 4), and (b) visualization system integrating a data-parallel visualization pipeline running on distributed memory systems with scalable remote rendering and immersive virtual environments (Section 5). Next, we continue with an example of scientific visualization of a continuously variable transmission dynamics in CAVE (Section 6). Finally we describe functionality of the augmented reality system developed by the part of our team at the High-Performance Computing Center, Stuttgart (Section 7).

2 Virtual engineering environment

Virtual environment systems are called decision making centers; they can be considered as a functional component of the virtual engineering workflow. The virtual engineering technologies in turn are extensions of digital engineering methods. Virtual prototyping technologies include: building geometric models in CAD/CADG systems; running high-performance Computer-Aided Engineering (CAE)/Computational Fluid Dynamics (CFD)/Fluid-Structure Interaction (FSI) simulations by groups of users in distributed mode (collaborative work); high performance visualization, in CAVEs or on stereo display walls (Figure 1).

Supercomputing centers at Stuttgart and Aachen Universities have made great progress in the development of virtual engineering technologies [1-3]. Besides developing hardware components for virtual environment systems, they have designed modelling and visualization software packages COVISE (COllaborative VISualization and Simulation Environment) [4] and Virtual Reality for Scientific Technical Applications (ViSTA) [5]. Both packages are provided as open-source under the GNU Lesser General Public License.

Visualization tools are nowadays extensively enhanced for better interaction between the user and a virtual environment: the user operates with a virtual model of an object in a natural way similar to interaction with a real object (Figure 2a,b, Figure 3a,b). Functionality of VE input systems is much more complex than that of standard PCs: it includes optical systems measuring user motion by tracking positions and rotations of special markers (e.g. a marker suit, stereo glasses, a flystick or fingertracking devices).

Scientific visualization systems have been developed for quite a long time [4,5,6]. Up-to-date VE systems support multi-user mode, with concurrent input from two and more users. Communications between several clients and support for simultaneous work with several Dtrack Flysticks or Dtrack Fingertracking devices allow a team of designers/engineers/managers to analyze models and make important decisions in cooperation.

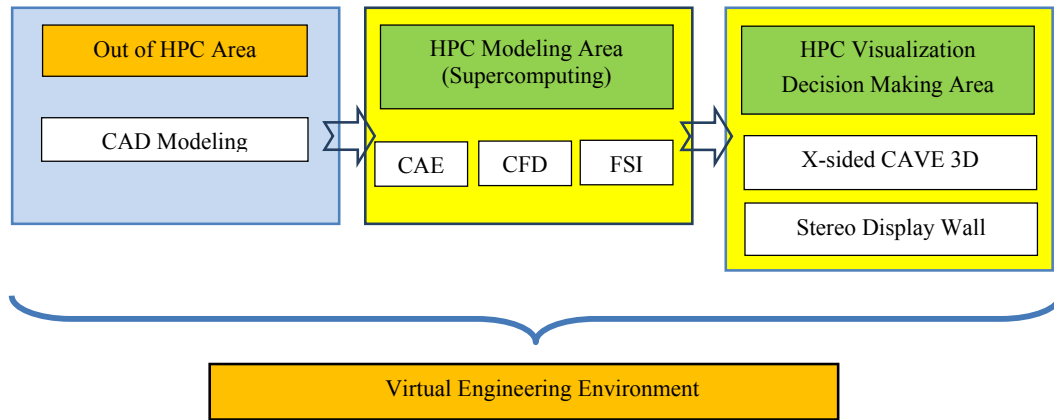


Figure 1. Virtual Engineering Environment

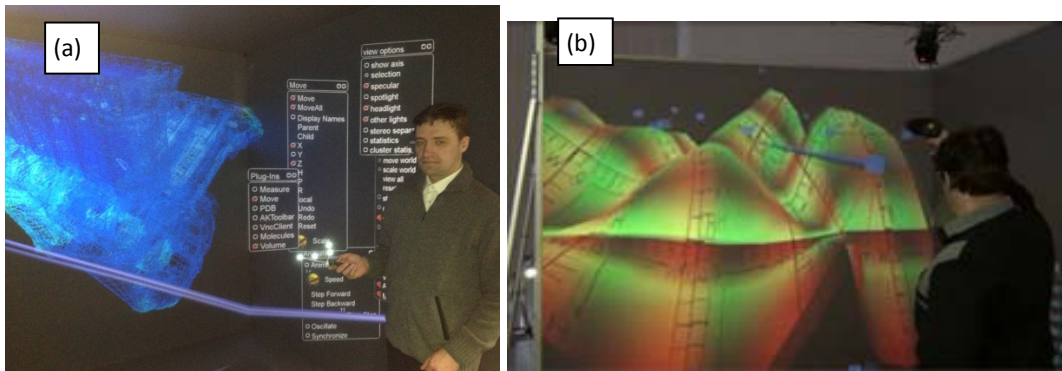


Figure 2. Interaction of a user with the virtual environment: (a) picking menu items with a flystick; (b) interactive morphing of a surface in real time

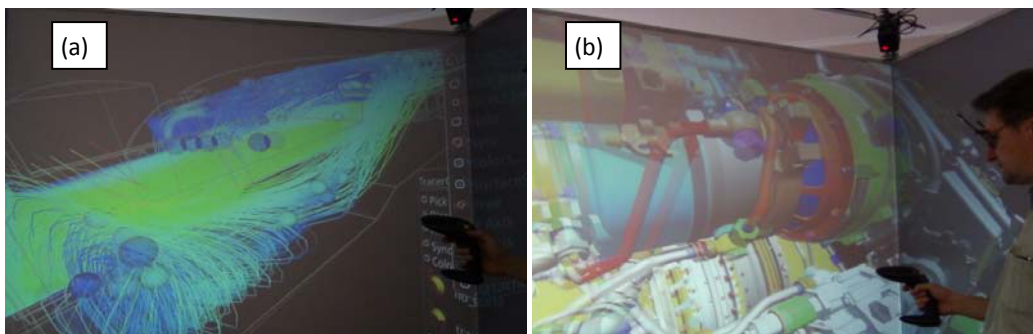


Figure 3. Working in a 3-sided CAVE 3D (St. Petersburg Polytechnic University): (a) Visualization of CFD analysis results; (b) Checking a CAD model to fix possible geometry faults

3 COVISE software system for scientific visualization in VEs

For visualization in CAVEs, we have used the software package COVISE [4] developed by the part of our team at the High Performance Computing Centre Stuttgart (HLRS), Germany. Lately we have extended the package in order to enhance user-environment interaction functionality. In the improved input system, new levels of abstraction have been introduced (Figure 4). This gives users and developers new functional possibilities, including:

- Co-working and communication of several users analyzing one virtual model;
- No limitations on usage of new input devices;
- An option to easily add new plug-ins supporting new input devices.

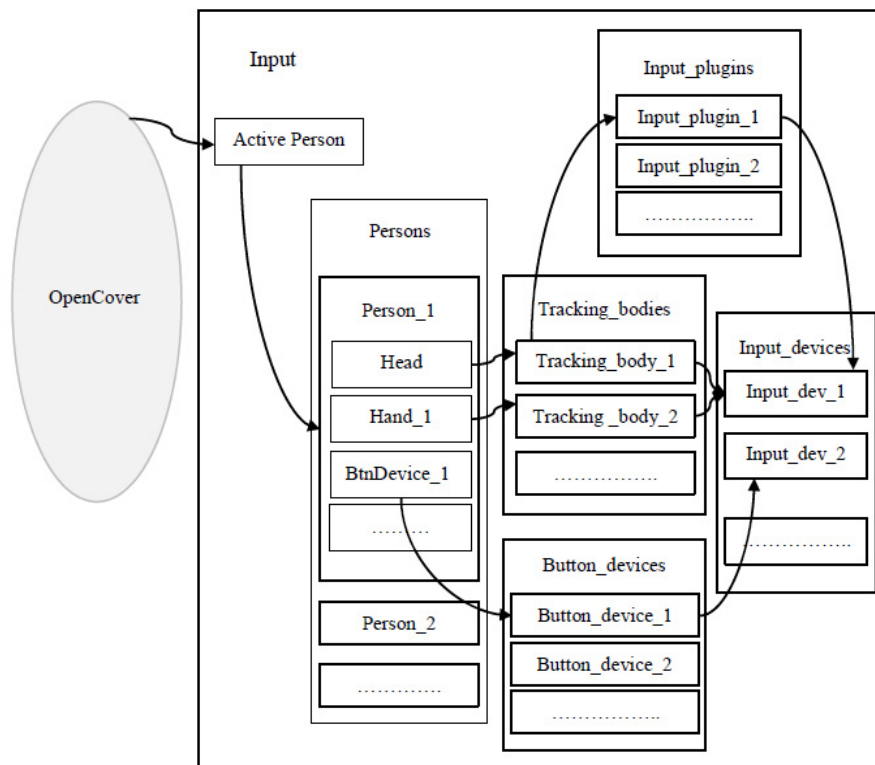


Figure 4. COVISE input system for OpenCOVER

4 Interactive generation and rendering of isosurfaces on large meshes

An actual problem in large-scale systems analyses is real-time visualization of non-stationary variable fields on meshes containing up to 10^{10} nodes. Solving this problem requires redesigning parallel volume rendering algorithms as well as methods of isosurface generation [9].

For interactive generation of an isosurface, a user must be able to change isosurface levels and color mapping in real-time. This requirement imposes limitations on the response time of visualization

tools: the new image should be generated within a period of 1-2 seconds, which requires adopting rendering algorithms for efficient work with large meshes. Below we briefly describe our visualization methods of generating isosurfaces on large meshes. A more detailed information on this work can be found in [7-8], along with the efficiency tests.

The considered non-stationary variable field is defined as a sequence of fields specified at discrete moments in time. The fields are defined on an unstructured tetrahedral mesh which, in general, does not fit into the main memory of one computational node. Therefore, the mesh is partitioned into a set of domains, where each domain contains an unstructured mesh of about 10^6 nodes. Nodes and elements are numbered locally in the domains; there is no need for global numeration as each domain is stored and processed on a separate computational node. As a result, a piece of the reduced isosurface is obtained in each domain; then the pieces are seamed pairwise in a loop. The implementation uses MPI and CUDA parallel programming tools [7].

A specific feature of CFD problems is overlapping domain boundaries with non-matching data in the intersections. As a result, two isosurfaces can overlap each other in the intersection region. A possible solution of this problem is merging two initial 3D tetrahedral meshes by, for example, mutual intersection of the initial meshes. This operation is simple to implement, however, it can result in a large number of new tetrahedrons, and many of them highly distorted. An alternative approach is independent generation of isosurfaces in each of the domains and subsequent merging of the isosurfaces based on topological connectivity and geometrical proximity of mesh elements. In this approach, a generalized pair contraction method [11] can be of much use.

The next stage of isosurface generation is isosurface extraction. For hexagonal meshes, the classical marching cubes approach is traditionally popular. However, it results in a large number of sliver triangles. The total number of triangles gets very high in this case, leading to increased rendering time and memory demand. In other approaches, such as dual marching cubes, total number of resulting triangles is less and their quality is higher, but approximation accuracy of the initial isosurface decreases. Marching tetrahedra is a generalized method for all types of unstructured meshes; however, it generates large numbers of sliver triangles, too.

The regularized marching tetrahedra method [12] and marching diamonds [13] overcome the drawbacks mentioned above. The resulting isosurface is always a manifold [14]. This will be true when generating isosurfaces for a field of unique values. If a field has the same value within some volume, this situation can be excluded by using the approach proposed in [10]. If the field value in a mesh node equals the level of the isosurface, we give a small increment to the node value, so that visualization realism is preserved but the isosurface does not contain the node position.

In general case, the resulting isosurface contains too many triangles and requires mesh reduction which is the next visualization stage. One of the main problems at this stage is the high memory demand during the initial reduction of the mesh. This problem is solved by using out-of-core algorithms for mesh reduction, especially stream reduction. The approach implemented in this work has been described in [7]; it employs the edge contraction algorithm (Figure 5a). The term “edge contraction” was proposed in [15].

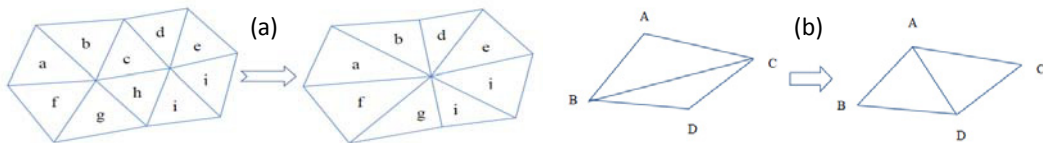


Figure 5. Mesh reduction and improvement: (a) edge contraction; (b) edge flipping

However, pair contraction in its general form leads to non-manifold meshes and hence is not used. An additional operation of edge flipping has been proposed by us in [7]; it often helps to improve the quality of triangles (Figure 5b).

Algorithms of interactive visualization are being extensively developed nowadays; however, we can make some intermediate conclusions. Thus, the mesh reduction method described in [10] has proven to be an efficient tool for reducing relatively smooth surfaces. However, when simplifying surfaces of high topological complexity, which are typical in the problems of turbulent flow analysis, the method often fails to delete the desired number of edges.

5 Alternative approach: scalable hybrid remote visualization

As described before, handling of large data within COVISE can be achieved by grafting some parallel execution steps onto inherently serial workflows. However, the resulting system is not very well-integrated and does not extend easily to other visualization tasks. To address this, we are building Vistle [24], a system integrating a data-parallel visualization pipeline running on distributed memory systems with scalable remote rendering and immersive virtual environments such as CAVEs. The guiding ideas in the design of the system are:

Reuse domain decompositions from simulations This allows to distribute the post-processing data and work load across the nodes of a cluster, and allows scaling visualization algorithms with the problem size. Typically, visualization algorithms run fast compared to the simulation. Thus, this approach yields enough parallelism and the visualization steps do not benefit from better load balancing through data redistribution.

Scalable remote rendering In order to handle arbitrary sizes of simulation data, we designed the system to be able to make use of the computer used during simulation runs. Often, these compute systems do not contain GPUs. Therefore, we implemented a render module running purely on CPUs based on the Embree [25] ray-tracing framework. Scalability is reached by a hybrid sort-first/sort-last approach: on each node, local data is rendered in parallel for several tiles of the output image. Depth compositing with IceT [26] is used for combining the images from all nodes into a complete result containing the contributions from all domain blocks.

Decouple interaction from remote rendering Immersive virtual environments require low interaction latencies: for being able to control the system, for modifying parameters of the visualization algorithms (such as isovalue), for continuously adapting the rendering to the current position of the user's head. We achieve this decoupling by a hybrid remote rendering [27] approach: menus, interaction elements for controlling visualization parameters (e.g. handles for cutting surfaces), and context geometry (e.g. input geometry to the simulation from the CAD system) are rendered locally at high frame rates. Images from the scalable remote rendering are overlaid asynchronously with the local rendering. Similar to sort-last parallel rendering, taking into account the depth values of local and remote images ensures that compositing yields a correct image. In order to hide remote rendering latencies even further, the remote depth data can be used to re-project the remote pixels according to updated viewer positions.

While this hybrid remote visualization approach is viable if a CAVE is connected to a local supercomputer with a fast network (e.g. 10GigE), it was shown that this strategy also works over long distances during last year's Supercomputing in New Orleans (USA) at the HLRS booth. Post-processing and rendering took place on a cluster at HLRS in Stuttgart, Germany. The results have been displayed on a stereo 3D display with 1400x1050 pixels (2.94 MPix per stereo frame) with head tracking at the booth. Interaction was smooth due to high local display update rates. Placing cutting surfaces and changing the isovalue was possible from within the virtual environment. After less than a second, updated images for the new parameters have become available, even though network round-trip times to Stuttgart were about 200 ms. A shared network connection to Stuttgart was used. The available bandwidth was about 10 MB/s. With full compression, display updates occurred at rates of about 10 frames/s.

6 Application example: visualization of continuously variable transmission dynamics in a CAVE

The example describes visualization of continuously variable transmission (CVT) dynamics. Our CVT model is original; it has been published in [16]. A specific feature of this CVT model is the combination of relatively small problem sizes (about 1000 unknowns) and high computational costs (up to several weeks of sequential code computations for a thirty seconds simulation period). The main source of computational complexity here is the calculation of non-linear contact forces acting between transmission parts at each step of the explicit time integration procedure.

Numerical simulations of CVT dynamics are commonly employed in the engineering practice at the design stage. However, at the high level of model complexity and realism, simulations become very time-consuming. Depending on the model complexity, it can take from several weeks to several months of sequential code computations for a thirty seconds model time period. A number of CVT models with different levels of realism and complexity have been proposed, including: (a) primitive low-dimension chain models [17]; (b) discrete multi-body mechanical models [18,19]; (c) continuous FEM models [20]. Our CVT model has been specially designed in order to keep an optimal balance between model realism and computational costs. The model employs the mathematical formalism of Lagrangian mechanics [22]; it is described by a system of ordinary differential equations representing the dynamics of a discrete system of elastic bodies subjected to contact, inertial and frictional forces. Reliability of the model has been validated during more than ten years of successful application in a real industrial environment [16].

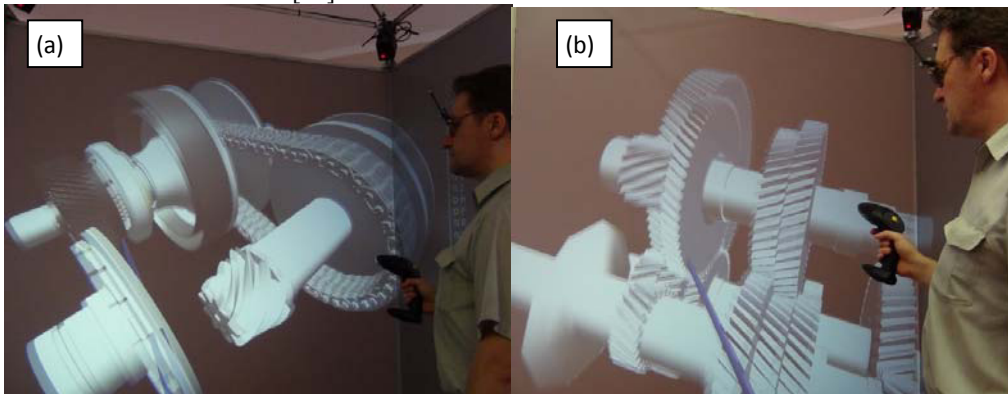


Figure 6. Continuously variable transmission analysis in a CAVE: (b) visualization of the CAD model; (a) visualization of simulated CVT dynamics

A parallel implementation of the CVT model has naturally become our next step in building a really fast application. At the current working stage, a task-based OpenMP parallelization has been performed for the most time-consuming portion of the code. Parallel performance analysis has been published in [23]. Figure 6 shows a visualization of the CVT model in CAVE: (a) checking a CAD model; (b) animation of CVT dynamics.

7 Scientific visualization in augmented reality

Physical testbeds such as wind tunnels or engine test benches still play an important role in product development today. The simulation models used for virtual prototyping need to be verified against physical tests, boundary conditions for simulations are often acquired in test benches and some aspects

just cannot be simulated today, therefore physical tests will always be a part of engineering workflows. By using augmented reality (AR), we can now combine physical tests with virtual prototypes in one environment. Figure 8 shows an application where the airflow inside a fill-finish production line for vials and syringes is visualized inside the real machine. Traditionally, smoke studies are carried out during the installation at the client's factory. Issues that appear during these tests have to be corrected on site, which is time consuming and expensive.

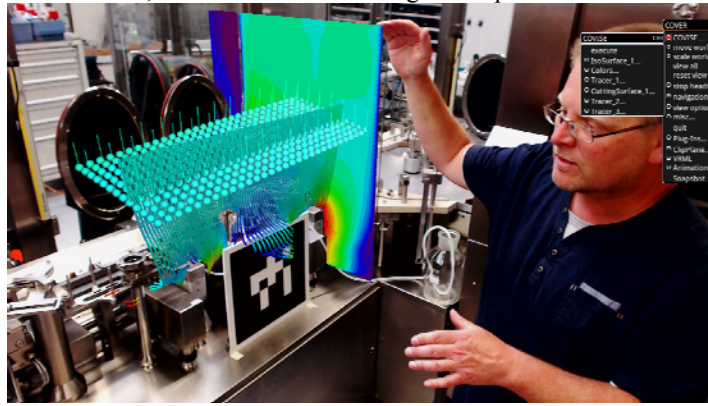


Figure 7. AR visualization of airflow through a Lino Line (OptimaPharma)

By using this augmented reality visualization, workers in the workshop can already take into account airflow phenomena and thus position particle counters or agar plates more efficiently, optimize the location of manipulation gloves and modify air inlets or outlets accordingly.

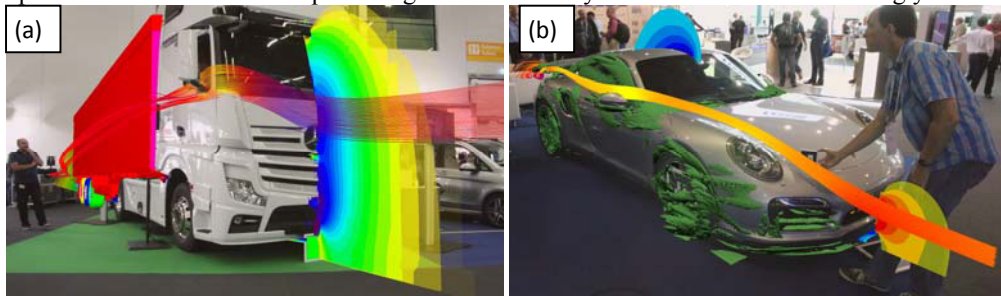


Figure 8. AR airflow visualization: (a) Velocity field and trailer on a Mercedes-Benz Actros; (b) Velocity field, pressure and helicity (green surface) on a Porsche 911 Turbo S

Augmented reality can also be used in wind tunnel experiments to compare simulated values with the real experiment. Figure 8a shows a pressure distribution on a longitudinal cut through the simulated dataset. This can then be directly compared to measured surface pressure at these positions during a wind tunnel experiment. While pressure measurements can only be carried out on the car surface, the augmented display can also show data values in 3D space, which gives the engineer a better context and more information than he normally would have during the physical experiment. Similarly, the user can display derived values such as helicity, a vortex criterion (Figure 8b) which also provides additional information to the experiment. For this experiment, we extended the COVISE system by a plug-in which detects fiducial markers and starts a configurable number of particles from this position which gives a virtual smoke wand (Figure 8b) with which the user can easily explore the airflow all around the car. It's also possible to attach that marker to a physical smoke wand and thus directly compare the simulated particle traces to real smoke traces during an experiment.

8 Conclusions

Powerful visualization of large-scale models is extremely important for the future of scientific research. In this paper, we have described our experience of using virtual environment and augmented reality for visualization of complex CFD and Lagrangian mechanics problems.

We have proposed and developed several software extensions aimed to improve visualization performance for large-scale models in 3D virtual environment systems such as CAVEs and Powerwalls. The first extension is an algorithm for real-time generation of isosurfaces on large meshes of up to 10^{10} nodes. The method has proven to be an efficient tool for reducing relatively smooth surfaces. However, when simplifying surfaces with topological complexity, the method often failed to delete the desired number of edges and highly faceted isosurfaces result. Our future plans include enhancement of the method for highly faceted surfaces.

Another software extension presented in the paper is the visualization system Vistle, integrating a data-parallel visualization pipeline running on massively parallel systems with scalable remote rendering and immersive virtual environments. This hybrid remote visualization approach has been successfully tested, providing display updates at rates of about 10 frames per second on a stereo 3D display with resolution of 2.94 MPix per stereo frame.

9 Acknowledgements

The research has been carried under the frame of the projects RBRF # 13-07-12077 ofi-m; CRESTA, which has received funding from the European Community's Seventh Framework Programme (ICT-2011.9.13); and bwVisu, which is funded by Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (MWK).

References

1. Wössner U., Kieferle J., Becker M. Interactive Simulations - An Intuitive Way of Making Supercomputing Ressources Available to the End User. Aktuelle Trends in der Softwareforschung, pp.84-94, do it.software-forschungstag, 2007.
2. M. Becker and U. Wössner (2005). Tangible interfaces for interactive flow simulation, 2nd Russian-German Advanced Research Workshop on Computational Science and High Performance Computing, March 14 - 16, 2005.
3. Research Challenges for Visualization Software. Childs, H. ; Geveci, B. ; Schroeder, W. ; Meredith, J. ; Moreland, K. ; Sewell, C. ; Kuhlen, T. ; Bethel, E.W. Computer, 2013 , Page(s): 34 - 42
4. COLlaborative VISualization and Simulation Environment (COVISE) homepage <http://www.hlr.de/organization/av/vis/covise/>
5. Virtual Reality for Scientific Technical Applications (ViSTA) homepage <http://www.itc.rwth-aachen.de/go/id/fgmo>
6. T.A. DeFanti, G. Dawe, D.J. Sandin, J.P. Schulze, P. Otto, J. Girado, F. Kuester, L. Smarr, R. Rao, "TheStarCAVE, a third-generation CAVE and virtual reality OptIPortal", Future Generation Computer Systems 25(2), Elsevier, February 2009, pp. 169-178.
7. Akayev A.A, Kuzin A.K., Orlov S.G., Chetverushkin B.N., Shabrov N.N., Iakobovski M.V. Generation of isosurface on Large Mesh. Proceeding of the IASTED International Conference ACIT 2010 in cooperation with Russian Academy of Science, page 236-240.

8. Shabrov N. N., Kuzin A. K., Orlov S. G., Chetverushkin B. N., Iakobovski M. V., Virtual prototyping modeling in the cave 3D environment, ICAS Congress, St. Petersburg, Sept. 2014.
9. S.V. Polyakov, M.V. Yakobovsky. Geometric modelling and visualization in the problems of modern electronics (in Russian). "Scientific Visualization", 2009, 1(1), c. 19-65.
10. A. Alyoshin., V. Afanasiev, P. Brusentsev, Ye. Yeremchenko., A. Klimenko, S. Klimenko, I. Nikitin, L. Nikitina, V. Pestrikov, A. Surin, O. Surina. Actual information technologies: scientific visualization, virtual environment, neo-geography, tangible images (in Russian). //Scientific visulalization (electronic journal). v. 5, № 4, p.117, 2013.
11. P. Borodin, S. Gumhold, M. Guthe, R. Klein. High-quality simplification with generalized pair contractions. In GraphiCon 2003, September 2003
12. G. M. Treece, R. W. Prager, and A. H. Gee. Regularised marching tetrahedra: improved isosurface extraction. *Comput. Graph.*, 23(4):583–598, 1999.
13. Anderson J.C., Bennett J. and Joy K.I. Marching Diamonds for Unstructured Meshes. *IEEE Visualization 2005*, pp 423–429, 2005.
14. D. P. Luebke. A developer's survey of polygonal simplification algorithms. *IEEE Comput. Graph.*, 21, 2001.
15. Garland M. and Heckbert P. S. Surface simplification using quadric error metrics. *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, pp 209–216, 1997.
16. N. Shabrov, Yu. Ispolov, S. Orlov. Simulations of continuously variable transmission dynamics. *Journal of Applied Mathematics and Mechanics* 94(11), pp. 917-922. WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim. DOI 10.1002/zamm.201300249 (2014).
17. M. Bullinger, F. Pfeiffer, and H. Ulbrich, Elastic modelling of bodies and contacts in continuous variable transmissions, *Multibody Syst. Dyn.* 13, 175–194 (2005).
18. L. Neumann, H. Ulbrich, and F. Pfeiffer, New model of a CVT rocker-pin chain with exact joint kinematics, *J. Comput. Nonlinear Dyn.* 1(2), 143–149 (2006).
19. L. Neumann, H. Ulbrich, and F. Pfeiffer, Optimization of the joint geometry of a rocker pin chain, *Machine Dyn. Problems* 29(4), 97–108 (2005).
20. M.T. Lates, R.G. Velicu, R. Papuc. Multiscale modeling of chain-guide contact by using tests and FEM. 11th World Congress on Computational Mechanics WCCM XI (2014).
21. Th. Geier, M. Foerg, R. Zander, and H. Ulbrich, F. Pfeiffer, A. Brandsma, A. van der Velde, Simulation of a push belt CVT considering uni- and bilateral constraints, *Journal of Applied Mathematics and Mechanics* 86(10), pp. 795–806 (2006).
22. V. Arnold, *Mathematical Methods of Classical Mechanics*, 2nd edition (Springer, New York, 1989), Chap. 3.
23. S.G. Orlov, N.B. Melnikova, Yu.G. Ispolov, N.N. Shabrov. High-performance simulations of continuously variable transmission dynamics (2015). Accepted for publication by Russian Supercomputing Days conference.
24. M. Aumüller, "The Architecture of Vistle, a Scalable Distributed Visualization System" in S. Markidis and E. Laure (Eds.): *Solving Software Challenges for Exascale*, EASC 2014, LNCS 8759, pp. 141–147, Springer, 2015.
25. I. Wald, S. Woop, C. Benthin, G. S. Johnson, and M. Ernst, "Embree: a kernel framework for efficient CPU ray tracing," *ACM TOG*, vol. 33, no. 4, pp. 143–8, Jul. 2014.
26. K. Moreland, W. Kendall, T. Peterka, and J. Huang, "An image compositing solution at scale," 2011, pp. 1–10.
27. C. Wagner, M. Flatken, F. Chen, A. Gerndt, C. D. Hansen, C. Hansen, and H. Hagen, "Interactive Hybrid Remote Rendering for Multi-pipe Powerwall Systems," in *Virtuelle und Erweiterte Realität - 9. Workshop der GI-Fachgruppe VR/AR*, C. Geiger, J. Herder, and T. Vierjahn, Eds. Aachen: Shaker Verlag, 2012, pp. 155–166.